WEST Search History

| Hide Items | Restore | Clear | Cancel |

DATE: Wednesday, August 02, 2006

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC; PLUR=NO; OP=OR	
☐	L36	L35 and (((character adj1 string$) or character-string$ or (text adj1 string$) or text-string$ or word or words) near match$)	1
☐	L35	(l26 or l27 or l28 or l29 or l30) and (xml near (document or documents) near (row or rows))	15
☐	L34	l31 and (xml near (document or documents) near (row or rows))	6
☐	L33	l31 and (xml near document near tree near (row or rows))	0
☐	L32	L31 and ((character adj1 string$) or character-string$ or (text adj1 string$) or text-string$ or word or words)	21
☐	L31	(l26 or l27 or l28 or l29 or L30) and (xml near document near tree near (node$ or parent or root or child$))	21
☐	L30	715/513.ccls.	2770
☐	L29	707/100.ccls.	4516
☐	L28	707/6.ccls.	1841
☐	L27	707/3.ccls.	7039
☐	L26	707/1.ccls.	5048
☐	L25	L22 and match$	1
☐	L24	L22 and (start or begin)	0
☐	L23	L22 and ((character adj1 string$) or character-string$ or (text adj1 string$) or text-string$ or word or words)	0
☐	L22	L21 and xml	1
☐	L21	L20 and (document or documents)	1
☐	L20	L19 and (row or rows)	1
☐	L19	20040044959.pn.	1
☐	L18	L17 and (end or finish)	1
☐	L17	L16 and (start or begin)	1
☐	L16	L15 and (row or rows)	1
☐	L15	L14 and ((character adj1 string$) or character-string$ or (text adj1 string$) or text-string$ or word or words)	1
☐	L14	L13 and match$	1
☐	L13	20030014397.pn.	1
☐	L12	(xml with (document or documents) with ((character adj1 string$) or character-string$ or (text adj1 string$) or text-string$ or word or words) near match$)	5

101 780, 404

☐	L11	(xml near (document or documents) near ((character adj1 string$) or character-string$ or (text adj1 string$) or text-string$ or word or words) near match$)	2
☐	L10	L9 and (tree near (document or documents))	17
☐	L9	L8 and ((character adj1 string$) or character-string or sting or strings or (text adj1 string) or text-string or word or words)	58
☐	L8	(xml near (document or documents) near match$)	76
☐	L7	(xml near (document or documents) near (row or rows))	21
☐	L6	L5 and ((xml adj1 (document or documents)) near tree)	15
☐	L5	(xml near document or documents near retriev$ near system).ti.	46
☐	L4	(xml near document or documents near retriev$ near system)	5749

DB=JPAB; PLUR=NO; OP=OR

| ☐ | L3 | (xml near document or documents near retriev$ near system) | 149 |
| ☐ | L2 | (xml near document or documents near retriev$) | 441 |

DB=USPT; PLUR=NO; OP=OR

| ☐ | L1 | (6635089 6519617 6557043 6571292 6671853 6681370 6781609 6904562 6938204 6941511 6463440 6523062 6662342 7069503 6397219 6466940 6487566 6886115 7043686 6094649 6366934 6421656 6519597 6584459 6606620 6625596 6826555 6832219 6898609 6981002 7043472 7062709 6635088 6832215 6931590 6966027 7020838 6707581 6810414 6901410 6476833 6745206 6847960 7054859 7058883 6182029 6199081 6249794 6738767 6850950).pn. (6067559 6088675 6125391 6223190 6226675 6223190 6226675 6279006 6301614 6321265 6393456 6418448 6426778 6446113 6507856 6529905 6532455 6535884 6538673 6542911 6542912 6569207 6589291 6604100 6613098 6631379 6636845 6640241 6643633 6654734 6654737 6657568 6675353 6681223 6684204 6684216 6684222 6684370 6711554 6721727 6725231 6732095 6748569 6763343 6763499 6766298 6766326 6766330 6772216 6782380).pn. (6785673 6785685 6785902 6792575 6792577 6799184 6804662 6804677 6823361 6826553 6829745 6836778 6836857 6845380 6845499 6850948 6859821 6871204 6874141 6874146 6883137 6904432 6907455 6910029 6912538 6915304 6920607 6920608 6928449 6928640 6931532 6934712 6934740 6938079 6941459 6941510 6947945 6948133 6950985 6959415 6959416 6961760 6968500 6971096 6978422 6981222 6986121 6990632 6993476 6993714).pn. (6993715 6996571 6996773 6996781 7007230 7007231 7013311 7013424 7017112 7020681 7020683 7024425 7031956 7047253 7051042 7055094 7058644 7062507 7062708 7069504 7072896 7073123 7076729 7055093 7051040 5469354 5821929 5748953 5845304 5465353 5649218 5842217 5892843 6003043 6144963 4873426 4985863 5047918 5220625 5265242 5327341 5457794 5590317 5628003 5680612 5706365 5706497 5752021 5787414 5860075).pn. (5940846 6035338 6108674 6169999 6278992 6327387 6437869 7010519 7047238 7072889 6263332 6701314 6779025 7013425 6490591 6898761 7054854 7058645 6311194 6449620 6539422 6581062 6684789 6741242 6764009 6792428 6810136 6810429 6817008 6826597 6854120 6857013 6886005 6964009 6996776 7003722 7016910 7024413 7035866 6083276 6502112 6601075 6240407 6249844 6405211 6442595 6480865 6490564 6507817 6507857).pn. (6578000 6591260 6604099 6643650 6658428 6675355 6717593 6725424 6725426 6769606 6779154 6796489 6799299 6807565 | 297 |

6823495 6825781 6829606 6829746 6847999 6880125 6886169 6889360
6895551 6901431 6901441 6907564 6910040 6912529 6915456 6925631
6934908 6940953 6947932 6950984 6952800 6952802 6954896 6963869
6964015 6976020 6986101 6990514 6990585 6990654 6996770 7010742
7013426 7016963 7020651).pn.

END OF SEARCH HISTORY

Dial g DataStar.

| options | logoff | feedback | help |

Advanced Search:

Inspec - 1898 to date (INZZ)

Search history:

No.	Database	Search term	Info added since	Results	
1	INZZ	xml NEAR (document OR documents) NEAR (row OR rows)	unrestricted	1	show titles
2	INZZ	xml NEAR (document OR documents)	unrestricted	2051	show titles
3	INZZ	2 AND tree	unrestricted	315	show titles
4	INZZ	3 AND start ADJ tag WITH end ADJ tag	unrestricted	0	-
5	INZZ	2 AND start ADJ tag WITH end ADJ tag	unrestricted	0	-
6	INZZ	3 AND tag OR tags	unrestricted	2755	show titles
7	INZZ	3 AND (tag OR tags)	unrestricted	15	show titles
8	INZZ	7 AND (character OR text) NEAR string$	unrestricted	0	-

hide | delete all search steps... | delete individual search steps...

Enter your search term(s): Search tips   ☐ Thesaurus mapping

[   ] [whole document   ▼] ◉

Information added since: [   ] or: [none   ▼]
(YYYYMMDD)

[search]

☐ Documents with images

Select special search terms from the following list(s):
⊙ Publication year 1950-
⊙ Publication year 1898-1949
⊙ Inspec thesaurus - browse headings A-G
⊙ Inspec thesaurus - browse headings H-Q
⊙ Inspec thesaurus - browse headings R-Z
⊙ Inspec thesaurus - enter a term

10\780,404

Dialog DataStar

options　　　logoff　　　feedback　　　help

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

locally as: PDF document　　　search strategy: do not include the search strategy

☑ Select All

1 SIOUX: an efficient index for processing structural XQueries.
2 XML query processing using a schema-based numbering scheme.
3 Prefix path streaming: a new clustering method for optimal holistic XML twig
4 Discovery of maximally frequent tag tree patterns with contractible variabl
5 An abstract grammar for XML document editing.
6 A new path expression computing approach for XML data.
7 Incremental validation of XML documents.
8 Naming in XML documents.
9 Efficient structural joins on indexed XML documents.
10 Discovery of frequent tag tree patterns in semi-structured Web documen
11 Discovery of frequent tree structured patterns in semistructured Web document
12 Adaptive conversion of Web content for mobile terminals.
13 Querying XML documents made easy: nearest concept queries.
14 An automated approach for retrieving hierarchical data from HTML tables.
15 An XML document to JavaScript object converter.

☑ **document 1 of 15** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update
　　0008680237 20051211.
Title
　　SIOUX: an efficient index for processing structural XQueries.
Conference information
　　Database and Expert Systems Applications. 16th International Conference, DEXA 2005. Proceedings, Copenhagen, Denmark, 22-26 Aug. 2005.
Source
　　Database and Expert Systems Applications. 16th International Conference, DEXA 2005. Proceedings (Lecture Notes in Computer Science Vol. 3588), 2005, p. 564-75, 22 refs, pp. xx+955, ISBN: 3-540-28566-0. Publisher: Springer-Verlag, Berlin, Germany.
Author(s)
　　Gardarin-G, Yeh-L.
　　Editor(s): Andersen-K-V, Debenham-J, Wagner-R.
Author affiliation
　　Gardarin, G., Yeh, L., PRiSM Lab., Univ. of Versailles, France.

Abstract
　　XML DBMSs require new indexing techniques to efficiently process structural search and full-text search as integrated in XQuery. Much research has been done for indexing **XML documents.** In this paper we first survey some of them and suggest a classification scheme. It appears that most

techniques are indexing on paths in **XML documents** and maintain a separated index on values. In some cases, the two indexes arc merged and/or **tags** are encoded. We propose a new method that indexes **XML documents** on ordered trees, i.e., two **documents** are in the same equivalence class is they have the same **tree** structure, with identical elements in order. We develop a simple benchmark to compare our method with two well-known European products. The results show that indexing on full trees leads to smaller index size and achieves 1 to 10 times better query performance in comparison with classical industrial methods that are path-based.

Descriptors
DATABASE-INDEXING;   QUERY-PROCESSING;   TREE-DATA-STRUCTURES;   XML.

Classification codes
C6160 Database-management-systems-DBMS*;
C6120 File-organisation;
C6130D Document-processing-techniques.

Keywords
SIOUX; structural-Xqueries-processing; full-text-search; **XML-document;** ordered-trees; **tree-structure;** query-performance.

Treatment codes
P Practical.

Language
English.

Publication type
Conference-proceedings.

Publication year
2005.

Publication date
20050000.

Edition
2005049.

Copyright statement
Copyright 2005 IEE.

☑ **document 2 of 15** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update
0008346623 20051201.

Title
XML query processing using a schema-based numbering scheme.

Conference information
Database and **XML** Technologies. Second International **XML** Database Symposium, XSym 2004. Proceedings, Toronto, Ont., Canada, 29-30 Aug. 2004.

Source
Database and **XML** Technologies. Second International **XML** Database Symposium, XSym 2004. Proceedings (Lecture Notes in Comput. Sci. Vol.3186), 2004, p. 21-34, 22 refs, pp. x+234, ISBN: 3-540-22969-8.
Publisher: Springer-Verlag, Berlin, Germany.

Author(s)
Kha-D-D, Yoshikawa-M.
Editor(s): Bellahsene-Z, Milo-T, Rys-M, Suciu-D, Unland-R.

Author affiliation
Kha, D.D., IMI Project of COE Program, Nagoya Univ.

Abstract
Establishing the hierarchical order among **XML** elements is an essential function of **XML** query processing techniques. Although most **XML documents** have an associated DTD or **XML** schema, the

document structure information has not been utilized efficiently in query processing techniques proposed so far. In this paper, we propose a novel technique that uses DTD or **XML** schema to improve the disk I/O complexity of **XML** query processing. We present a schema-based numbering scheme called SPIDER that incorporates both structure information and **tag** names extracted from the **document** structure descriptions. Given the **tag** name and the identifier of an element, SPIDER can determine the **tag** names and the identifiers of the ancestor elements without disk I/O. Based on SPIDER, we designed a mechanism called VirtualJoin that significantly reduces disk I/O workload for processing **XML** queries. Our experiments indicated that SPIDER outperforms the structural join techniques **Stack-Tree** and PathStack in **XML** query processing, especially for **XML** queries with heavy join workload and large data sets.

Descriptors

COMPUTATIONAL-COMPLEXITY; QUERY-PROCESSING; RELATIONAL-DATABASES; TREE-DATA-STRUCTURES; XML.

Classification codes

C6160D Relational-databases*;
C6130D Document-processing-techniques;
C4240C Computational-complexity;
C6120 File-organisation.

Keywords

XML-query-processing; schema-based-numbering-scheme; **XML-documents;** disk-I/O-complexity; SPIDER; **tag-names;** VirtualJoin; **Stack-Tree-** technique; PathStack-technique.

Treatment codes

P Practical.

Language

English.

Publication type

Conference-proceedings.

Publication year

2004.

Publication date

20040000.

Edition

2005013.

Copyright statement

Copyright 2005 IEE.

COPYRIGHT BY The IET, Stevenage, UK

☑ **document 3 of 15** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0008289033 20051201.

Title

Prefix path streaming: a new clustering method for optimal holistic **XML** twig pattern matching.

Conference information

Database and Expert Systems Applications. 15th International Conference, DEXA 2004. Proceedings, Zaragoza, Spain, 30 Aug.-1 Sept. 2004.

Source

Database and Expert Systems Applications. 15th International Conference, DEXA 2004. Proceedings (Lecture Notes in Comput. Sci. Vol.3180), 2004, p. 801-10, 8 refs, pp. xxi+972, ISBN: 3-540-22936-1.

Publisher: Springer-Verlag, Berlin, Germany.

Author(s)

Ting-Chen, Tok-Wang-Ling, Chee-Yong-Chan.
Editor(s): Galindo-F, Takizawa-M, Traunmuller-R.

Author affiliation
   Ting Chen, Tok Wang Ling, Chee-Yong Chan, Sch. of Comput., Nat. Univ. of Singapore.

Abstract
   Searching for all occurrences of a twig pattern in a **XML document** is an important operation in **XML** query processing. Recently a class of holistic twig pattern matching algorithms has been proposed. Compared with the prior approaches, the holistic method avoids generating large intermediate results which do not contribute to the final answer. The method is CPU and I/O optimal when twig patterns only have ancestor-descendant relationships. The holistic twig-pattern matching method proposed earlier (N. Bruno et al. (2002)) operates on element streams which cluster all **XML** elements with the same **tag** name together. In this paper we introduce a clustering method called prefix path streaming (PPS) and new holistic twig pattern matching algorithms based on PPS. PPS clusters elements of **XML documents** according to the paths from root to the elements. This clustering approach avoids unnecessary scanning of irrelevant portion of **XML documents.** More importantly, we develop optimal algorithms based on PPS streaming which can process a large class of twig patterns consisting of both ancestor-descendant and parent-child relationships.

Descriptors
   PATTERN-CLUSTERING; PATTERN-MATCHING; QUERY-PROCESSING; STATISTICAL-ANALYSIS; TREE-DATA-STRUCTURES; XML.

Classification codes
   C6130D Document-processing-techniques*;
   C6130M Multimedia;
   C1250 Pattern-recognition;
   C6120 File-organisation;
   C1140Z Other-topics-in-statistics;
   C6160 Database-management-systems-DBMS.

Keywords
   XML-document; XML-query-processing; holistic-twig-pattern-matching- algorithms; ancestor-descendant-relationships; clustering-method; prefix-path-streaming; parent-child-relationships.

Treatment codes
   P Practical.

Language
   English.

Publication type
   Conference-proceedings.

Publication year
   2004.

Publication date
   20040000.

Edition
   2005007.

Copyright statement
   Copyright 2005 IEE.

☑ **document 4 of 15** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update
   0008212238 20051201.

Title
   Discovery of maximally frequent **tag tree** patterns with contractible variables from semistructured **documents.**

Conference information
   Advances in Knowledge Discovery and Data Mining. 8th Pacific-Asia Conference, PAKDD 2004. Proceedings, Sydney, NSW, Australia, 26-28 May 2004.

Sponsor(s): SAS; Univ of Technol, Sydney.
Source
Advances in Knowledge Discovery and Data Mining. 8th Pacific-Asia Conference, PAKDD 2004. Proceedings (Lecture Notes in Artificial Intelligence Vol.3056), 2004, p. 133-44, 12 refs, pp. xix+713, ISBN: 3-540-22064-X.
Publisher: Springer-Verlag, Berlin, Germany.
Author(s)
Miyahara-T, Suzuki-Y, Shoudai-T, Uchida-T, Takahashi-K, Ueda-H.
Editor(s): Dai-H, Srikant-R, Zhang-C.
Author affiliation
Miyahara, T., Fac. of Sci., Hiroshima City Univ., Japan.

Abstract
In order to extract meaningful and hidden knowledge from semistructured **documents** such as HTML or **XML** files, methods for discovering frequent patterns or common characteristics in semistructured **documents** have been more and more important. We propose new methods for discovering maximally frequent **tree** structured patterns in semistructured Web **documents** by using **tag tree** patterns as hypotheses. A **tag tree** pattern is an edge labeled **tree** which has ordered or unordered children and structured variables. An edge label is a **tag** or a keyword in such Web **documents,** and a variable can match an arbitrary subtree, which represents a field of a semistructured **document.** As a special case, a contractible variable can match an empty subtree, which represents a missing field in a semistructured **document.** Since semistructured **documents** have irregularities such as missing fields, a **tag tree** pattern with contractible variables is suited for representing **tree** structured patterns in such semistructured **documents.** First, we present an algorithm for generating all maximally frequent ordered **tag tree** patterns with contractible variables. Second, we give an algorithm for generating all maximally frequent unordered **tag tree** patterns with contractible variables.

Descriptors
DATA-MINING; DOCUMENT-HANDLING; INTERNET; PATTERN-CLASSIFICATION; TREE-DATA-STRUCTURES; XML.
Classification codes
C6170K Knowledge-engineering-techniques*;
C6130D Document-processing-techniques;
C7210N Information-networks;
C6120 File-organisation.
Keywords
maximally-frequent-tag-tree-pattern-discovery; contractible-variable; HTML; **XML;** **semistructured-Web-document; edge-labeled-tree;** maximally- **frequent-unordered-tag-tree-pattern.**
Treatment codes
P Practical.
Language
English.
Publication type
Conference-proceedings.
Publication year
2004.
Publication date
20040000.
Edition
2004049.

---

document 5 of 15 Order Document
Inspec - 1898 to date (INZZ)

Accession number & update
> 0007822103 20051201.

Title
> An abstract grammar for **XML document** editing.

Source
> Journal of KISS: Software and Applications, {J-KISS-Softw-Appl-South-Korea}, April 2003, vol. 30, no. 3-4, p. 268-77, 15 refs, CODEN: CKNBFV, ISSN: 1229-6848.
> Publisher: Korea Inf. Sci. Soc, South Korea.

Author(s)
> Kyoun-Hee-Shin, Jong-Myung-Choi, Chae-Woo-Yoo.

Abstract

> A **document** type definition (DTD) which defines **tags** for a **document** is an **XML document** grammar that defines syntactic structure of a **document**. An **XML document** keeps the rules and must be parsed to check validation. To parse **XML document,** the deterministic parsing method of programming language is irrelevant because it does not satisfy the definition of deterministic content model in element declaration. In this paper, we consider editing of a valid **XML document** in syntax-directed editing environment, and we suggest the internal storage representations of syntax in DTD and their algorithms. The consequence is that a syntactic structure of textual DTD is transformed into graph and table structures. The table structure of DTD is interpreted as a context free grammar which has attribute values and is used in syntax-directed editor for **XML.** We called this the **XML** abstract grammar and showed generated results and examples.

Descriptors
> ATTRIBUTE-GRAMMARS; CONTEXT-FREE-GRAMMARS; DOCUMENT-HANDLING; PROGRAMMING-LANGUAGE-SEMANTICS; TREE-DATA-STRUCTURES; XML.

Classification codes
> C4210L Formal-languages-and-computational-linguistics*;
> C6130D Document-processing-techniques;
> C6140D High-level-languages;
> C6120 File-organisation.

Keywords
> abstract-grammar; **XML-document-editing;** syntax-directed-editing; **document-type-definition;** DTD; deterministic-parsing; syntactic- structure; internal-storage-representation; graph-structure; table-structure; reference-attribute; context-free-grammar.

Treatment codes
> P Practical.

Language
> Chinese.

Publication type
> Journal-paper.

Availability
> SICI: 1229-6848(200304)30:3/4L.268:AGDE; 1-W.

Publication year
> 2003.

Publication date
> 20030400.

Edition
> 2004001.

Copyright statement
> Copyright 2004 IEE.

☑ **document 6 of 15** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update

0007816121 20051201.

Title

A new path expression computing approach for **XML** data.

Conference information

Efficiency and Effectiveness of **XML** Tools and Techniques and Data Integration over the Web. VLDB 2002 Workshop EEXTT and CAiSE 2002 Workshop DIWeb. Revised Papers, London, UK, Dec. 2002.

Source

Efficiency and Effectiveness of **XML** Tools and Techniques and Data Integration over the Web. VLDB 2002 Workshop EEXTT and CAiSE 2002 Workshop DIWeb. Revised (Lecture Notes in Computer Science Vol.2590), 2003, p. 35-46, 11 refs, pp. x+258, ISBN: 3-540-00736-9.

Publisher: Springer-Verlag, Berlin, Germany.

Author(s)

Jianhua-Lv, Guoren-Wang, Jeffrey-Xu-Yu, Ge-Yu, Hongjun-Lu, Bing-Sun.

Editor(s): Bressan-S, Chaudhri-A-B, Lee-M-L, Yu-J-X, Lacroix-Z.

Author affiliation

Jianhua Lv, Guoren Wang, Northeastern Univ. of China, Shenyang, China.

Abstract

Most query languages in **XML** database systems use regular path expressions (RPE) to query or extract data from databases and some query processing and optimization techniques have been proposed for RPEs. Conceptually **XML documents** are collections of path instances. Each path instance should conform to an **XML** element **tag** sequence, called path schema. A RPE query can be written as an automaton that can represent a language, while path schemas can be seen as sentences. A novel RPE computing approach, automaton match (AM), is proposed. AM queries the RPEs by matching the automatons with path schemas. The experimental results show AM is quite efficient for computing RPE queries.

Descriptors

DATABASE-MANAGEMENT-SYSTEMS; FINITE-STATE-MACHINES; HYPERMEDIA-MARKUP-LANGUAGES; QUERY-LANGUAGES; QUERY-PROCESSING; TREE-DATA-STRUCTURES.

Classification codes

C6160 Database-management-systems-DBMS*;

C6130M Multimedia;

C4220 Automata-theory;

C6120 File-organisation;

C6130D Document-processing-techniques.

Keywords

query-language; **XML-database-system**; regular-path-expression; RPE-query; data-querying; data-extraction; query-processing-technique; query-optimization-technique; **XML-document; XML-element-tag-sequence;** path-schema; automaton-match.

Treatment codes

P Practical;

T Theoretical-or-mathematical.

Language

English.

Publication type

Conference-proceedings.

Publication year

2003.

Publication date

20030000.

Edition

2003050.

Copyright statement

Copyright 2003 IEE.

☑ **document 7 of 15** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update
0007816047 20051201.
Title
Incremental validation of **XML documents.**
Conference information
Database Theory - ICDT 2003. 9th International Conference. Proceedings, Siena, Italy, 8-10 Jan. 2003.
Source
Database Theory - ICDT 2003. 9th International Conference. Proceedings (Lecture Notes in Computer Science Vol.2572), 2003, p. 47-63, 25 refs, pp. xi+454, ISBN: 3-540-00323-1.
Publisher: Springer-Verlag, Berlin, Germany.
Author(s)
Papakonstantinou-Y, Vianu-V.
Editor(s): Calvanese-D, Lenzerini-M, Motwani-R.
Author affiliation
Papakonstantinou, Y., Vianu, V., Comput. Sci. & Eng., California Univ., San Diego, CA, USA.

Abstract
We investigate the incremental validation of **XML documents** with respect to DTDs and **XML** schemas, under updates consisting of element **tag** renamings, insertions and deletions. DTDs are modeled as extended context-free grammars and **XML** schemas are abstracted as "specialized DTDs", allowing to decouple element types from element **tags.** For DTDs, we exhibit an O(m log n) incremental validation algorithm using an auxiliary structure of size O(n), where n is the size of the **document** and m the number of updates. For specialized DTDs, we provide an O(m logof size O(n). This is a significant improvement over brute-force revalidation from scratch.

Descriptors
COMPUTATIONAL-COMPLEXITY; CONTEXT-FREE-GRAMMARS; DATA-MODELS; DATABASE-THEORY; FINITE-AUTOMATA; FORMAL-VERIFICATION; HYPERMEDIA-MARKUP-LANGUAGES; TREE-DATA-STRUCTURES.
Classification codes
C6130D Document-processing-techniques*;
C6130M Multimedia;
C4210L Formal-languages-and-computational-linguistics;
C4240C Computational-complexity;
C4220 Automata-theory;
C6110F Formal-methods;
C6120 File-organisation;
C4250 Database-theory;
C6160 Database-management-systems-DBMS.
Keywords
XML-document-incremental-validation-algorithm; XML-schema; context- free-grammar; **document-type-definition;** DTD.
Treatment codes
T Theoretical-or-mathematical.
Language
English.
Publication type
Conference-proceedings.
Publication year
2003.
Publication date
20030000.
Edition
2003050.
Copyright statement
Copyright 2003 IEE.

☑ **document 8 of 15**

Inspec - 1898 to date (INZZ)

Accession number & update
> 0007678530 20051201.

Title
> Naming in **XML documents.**

Conference information
> On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE. Confederated
> International Conferences Proceedings, Irvine, CA, USA, Aug. 2002.
> Sponsor(s): Boeing, USA; OntoWeb, Netherlands; Telecoria Technol., USA.

Source
> On the Move to Meaningful Internet Systems 2002. CoopIS, DOA, and ODBASE. Confederated
> International Conferences CoopIS, DOA, and ODBASE 2002 Proceedings (Lecture Notes in Computer
> Science Vol.2519), 2002, p. 1287-303, 24 refs, pp. xxiii+1367, ISBN: 3-540-00106-9.
> Publisher: Springer-Verlag, Berlin, Germany.

Author(s)
> Lawrence-R.
> Editor(s): Meersman-R, Tari-Z.

Author affiliation
> Lawrence, R., Dept. of Comput. Sci., Iowa Univ., Iowa City, IA, USA.

Abstract
> **XML** is now an established standard for data communication and representation. There has been
> considerable work on **XML** querying, modeling, and type definition. However, one of the most
> important aspects of **XML,** standardized **tag** naming for conveying semantics, has been almost ignored
> by the research community. This paper argues that the naming aspects of **XML** are important to
> consider and presents a naming methodology for **XML tags** that captures increased context
> information. Using semantic **tag** names opens up the possibility of semantic querying of **XML**
> **documents,** which simplifies query formulation by reducing the reliance on path expressions. A
> semantic query facility allows **XML documents** with similar semantics, but organized using different
> DTDs, to be queried without modifying the original query formulation. Finally, we demonstrate an
> algorithm for converting semantic queries to structural queries by disambiguating incomplete path
> expressions.

Descriptors
> HYPERMEDIA-MARKUP-LANGUAGES; NAMING-SERVICES; QUERY-LANGUAGES;
> QUERY-PROCESSING; TREE-DATA-STRUCTURES.

Classification codes
> C6120 File-organisation*;
> C6130D Document-processing-techniques;
> C6140D High-level-languages;
> C7240 Information-analysis-and-indexing;
> C6160 Database-management-systems-DBMS.

Keywords
> data-communication; data-representation; **XML-querying; XML-modeling;** incomplete-path-
> expressions; **XML-type-definition; standardized-tag-** naming; semantics; semantic-queries;
> structural-queries.

Treatment codes
> P Practical.

Language
> English.

Publication type
> Conference-proceedings.

Publication year
> 2002.

Publication date
> 20020000.

Edition
> 2003026.

Copyright statement
        Copyright 2003 IEE.

COPYRIGHT BY The IET, Stevenage, UK

☑ **document 9 of 15** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update
        0007664124 20051201.
Title
        Efficient structural joins on indexed **XML documents.**
Conference information
        VLDB 2002: 28th International Conference on Very Large Databases, Hong Kong, China, 20-23 Aug.
        2002.
Source
        Proceedings of the Twenty-eighth International Conference on Very Large Data Bases, 2002, p. 263-
        74, 40 refs, pp. xxvi+1118, ISBN: 1-55860-869-9.
        Publisher: Morgan Kaufmann Publishers, San Francisco, CA, USA.
Author(s)
        Shu-Yao-Chien, Vagena-Z, Donghui-Zhang, Tsotras-V-J, Zaniolo-C.
        Editor(s): Bernstein-P-A, Ioannidia-Y-E, Ramakrishnan-R, Papadias-D.
Abstract
        Queries on **XML documents** typically combine selections on element contents, and, via path
        expressions, the structural relationships between tagged elements. Structural joins are used to find all
        pairs of elements satisfying the primitive structural relationships specified in the query, namely,
        parent-child and ancestor-descendant relationships. Efficient support for structural joins is thus the key
        to efficient implementations of **XML** queries. Recently proposed node numbering schemes enable the
        capturing of the **XML document** structure using traditional indices (such as B+-trees or R-trees). This
        paper proposes efficient structural join algorithms in the presence of **tag** indices. We first concentrate
        on using B+-trees and show how to expedite a structural join by avoiding collections of elements that
        do not participate in the join. We then introduce an enhancement (based on sibling pointers) that
        further improves performance. Such sibling pointers are easily implemented and dynamically
        maintainable. We also present a structural join algorithm that utilizes R-trees. An extensive
        experimental comparison shows that the **B+-tree** structural joins are more robust. Furthermore, they
        provide drastic improvement gains over the current state of the art.
Descriptors
        HYPERMEDIA-MARKUP-LANGUAGES; QUERY-PROCESSING; TREE-DATA-STRUCTURES.
Classification codes
        C6160 Database-management-systems-DBMS*;
        C6130D Document-processing-techniques;
        C6130M Multimedia;
        C6140D High-level-languages;
        C6120 File-organisation.
Keywords
        indexed-XML-documents; efficient-structural-joins; element-contents; path-expressions; structural-
        relationships; tagged-elements; parent-child-relationships; ancestor-descendant-relationships; B+-
        trees; R-trees; sibling-pointers.
Treatment codes
        P Practical.
Language
        English.
Publication type
        Conference-proceedings.
Publication year
        2002.

Publication date
       20020000.
Edition
       2003024.
Copyright statement
       Copyright 2003 IEE.

COPYRIGHT BY The IET, Stevenage, UK

☑ **document 10 of 15** <u>Order Document</u>
Inspec - 1898 to date (INZZ)

Accession number & update
       0007386738 20051201.
Title
       Discovery of frequent **tag tree** patterns in semi-structured Web **documents.**
Conference information
       Advances in Knowledge Discovery and Data Mining. 6th Pacific-Asia Conference, PAKDD 2002.
       Proceedings, Taipei, Taiwan, 6-8 May 2002.
Source
       Advances in Knowledge Discovery and Data Mining. 6th Pacific-Asia Conference, PAKDD 2002.
       Proceedings (Lecture Notes in Artificial Intelligence Vol.2336), 2002, p. 341-55, 11 refs, pp. xiii+568,
       ISBN: 3-540-43704-5.
       Publisher: Springer-Verlag, Berlin, Germany.
Author(s)
       <u>Miyahara-T</u>, <u>Suzuki-Y</u>, <u>Shoudai-T</u>, <u>Uchida-T</u>, <u>Takahashi-K</u>, <u>Ueda-H</u>.
       Editor(s): <u>Chen-M-S</u>, <u>Yu-P-S</u>, <u>Liu-B</u>.
Author affiliation
       Miyahara, T., Fac. of Inf. Sci., Hiroshima City Univ.
Abstract
       Many Web **documents** such as HTML files and **XML** files have no rigid structure and are called semi-
       structured data. In general, such semi-structured Web **documents** are represented by rooted trees
       with ordered children. We propose a method for discovering frequent **tree** structured patterns in semi-
       structured Web **documents** by using a **tag tree** pattern as a hypothesis. A **tag tree** pattern is an
       edge labeled **tree** with ordered children which has structured variables. An edge label is a **tag** or a
       keyword in such Web **documents,** and a variable can be substituted by an arbitrary **tree.** So a **tag
       tree** pattern is suited to representing **tree** structured patterns in such Web **documents.** First we show
       that it is hard to compute the optimum frequent **tag tree** pattern. So we present an algorithm for
       generating all maximally frequent **tag tree** patterns and give the correctness of it. Finally, we report
       some experimental results on our algorithm. Although this algorithm is not efficient, experiments show
       that we can extract characteristic **tree** structured patterns in those data.
Descriptors
       COMPUTATIONAL-COMPLEXITY;   DATA-MINING;   HYPERMEDIA-MARKUP-LANGUAGES;
       INFORMATION-RESOURCES;   INFORMATION-RETRIEVAL;   TREES-MATHEMATICS.
Classification codes
       C7250R Information-retrieval-techniques*;
       C6130D Document-processing-techniques;
       C7240 Information-analysis-and-indexing;
       C1160 Combinatorial-mathematics;
       C4240C Computational-complexity;
       C7210N Information-networks;
       C6160 Database-management-systems-DBMS.
Keywords
       frequent-tag-tree-patterns; semi-structured-Web-documents; HTML-files; **XML-files;** semi-
       structured-data; rooted-trees; ordered-children; edge- **labeled-tree;** edge-label; keyword; **tree-
       structured-patterns.**

Treatment codes
P Practical;
T Theoretical-or-mathematical.
Language
English.
Publication type
Conference-proceedings.
Publication year
2002.
Publication date
20020000.
Edition
2002037.
Copyright statement
Copyright 2002 IEE.

☑ **document 11 of 15** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update
0007128641 20010101.
Title
Discovery of frequent **tree** structured patterns in semistructured Web **documents.**
Conference information
Advances in Knowledge Discovery and Data Mining. 5th Pacific-Asia Conference, PAKDD 2001, Hong Kong, China, 16-18 April 2001.
Sponsor(s): SAS; ETNet; Hong Kong Pei Hua Educ. Found; IEEE Comput. Soc. Hong Kong Sect., Comput. Chapter; ACM Hong Kong; et al.
Source
Advances in Knowledge Discovery and Data Mining. 5th Pacific-Asia Conference, PAKDD 2001. Proceedings (Lecture Notes in Artificial Intelligence Vol.2035), 2001, p. 47-52, 10 refs, pp. xviii+596, ISBN: 3-540-41910-1.
Publisher: Springer-Verlag, Berlin, Germany.
Author(s)
Miyahara-T, Shoudai-T, Uchida-T, Takahashi-K, Ueda-H.
Editor(s): Cheung-D, Williams-G-J, Li-Q.
Author affiliation
Miyahara, T., Fac. of Inf. Sci., Hiroshima City Univ.
Abstract
Many **documents** such as Web **documents** or **XML** files have no rigid structure. Such semistructured **documents** have been rapidly increasing. We propose a new method for discovering frequent **tree** structured patterns in semistructured Web **documents.** We consider the data mining problem of finding all maximally frequent **tag tree** patterns in semistructured data such as Web **documents.** A **tag tree** pattern is an edge labeled **tree** which has hyperedges as variables. An edge label is a **tag** or a keyword in Web **documents,** and a variable can be substituted by any **tree.** So a **tag tree** pattern is suited for representing **tree** structured patterns in semistructured Web **documents.** We present an algorithm for finding all maximally frequent **tag tree** patterns. Also we report some experimental results on **XML documents** by using our algorithm.
Descriptors
DATA-MINING; INFORMATION-RESOURCES; TREE-DATA-STRUCTURES.
Classification codes
C7210N Information-networks*;
C6120 File-organisation;
C6170K Knowledge-engineering-techniques.

Keywords
> frequent-tree-structured-patterns-discovery; XML-files; data-mining; tag-tree-patterns; edge-labeled-tree; hyperedges; XML-documents; semistructured-Web-documents.

Treatment codes
> A Application;
> P Practical.

Language
> English.

Publication type
> Conference-proceedings.

Publication year
> 2001.

Publication date
> 20010000.

Edition
> 2001050.　　　　　,

Copyright statement
> Copyright 2001 IEE.

☑ **document 12 of 15** Order Document

Inspec - 1898 to date (INZZ)

Accession number & update
> 0007209687 20051201.

Title
> Adaptive conversion of Web content for mobile terminals.

Source
> Journal of KISS: Computing Practices, {J-KISS-Comput-Pract-South-Korea }, Dec. 2000, vol. 6, no. 6, p. 635-42, 14 refs, CODEN: CKNCFY, ISSN: 1229-7712.
> Publisher: Korea Inf. Sci. Soc, South Korea.

Author(s)
> Sueng-chun-Kang, Kwangsue-Chung.

Abstract
> In this paper, we propose an efficient **document** conversion mechanism to provide an adaptive Web **document** to mobile terminals. We also propose a RHTML (reduced HTML) to archive the adaptive **tag** reduction. A markup error correction process in the proposed adaptive **document** conversion mechanism converts an HTML (HyperText Markup Language) **document** into an **XML** (Extensible Markup Language) application **document.** This process makes Web **documents** easy to handle with DOM **(document** object model) as the **tree** model, and it removes the hardware overhead in mobile terminals. Also, a **tag** reduction process provides the adaptive Web **document** with three DTDs **(document** type definitions) in the RHTML.

Descriptors
> ADAPTIVE-SYSTEMS; DOCUMENT-HANDLING; ELECTRONIC-DATA-INTERCHANGE; ERROR-CORRECTION; HYPERMEDIA-MARKUP-LANGUAGES; INFORMATION-RESOURCES; INTERACTIVE-TERMINALS; MOBILE-COMPUTING; TREE-DATA-STRUCTURES.

Classification codes
> C6130D Document-processing-techniques*;
> C7210N Information-networks;
> C6130M Multimedia;
> C6130E Data-interchange;
> C6150N Distributed-systems-software.

Keywords
> adaptive-Web-content-conversion; mobile-terminals; **document-** conversion-mechanism; **adaptive-Web-document;** World-Wide-Web; reduced-HTML; **adaptive-tag-reduction-archiving;** markup-

error-correction-process; **XML-application-document; DOM; document-object-model; tree-model;** hardware-overhead; **document-type-definitions.**
Treatment codes
    P Practical.
Language
    Korean.
Publication type
    Journal-paper.
Availability
    SICI: 1229-7712(200012)6:6L.635:ACCM; 1-#.
Publication year
    2000.
Publication date
    20001200.
Edition
    2002011.
Copyright statement
    Copyright 2002 IEE.

☑ **document 13 of 15** Order Document
Inspec - 1898 to date (INZZ)

Accession number & update
    0006912587 20051201.
Title
    Querying **XML documents** made easy: nearest concept queries.
Conference information
    Proceedings of 17th IEEE International Conference on Data Engineering, Heidelberg, Germany, 2-6
    April 2001.
    Sponsor(s): IEEE Comput. Soc. Tech. Committee on Data Eng; EML; IBM; Hewlett-Packard; SAS;
    Microsoft; ABB; Software AG; sd&m.
Source
    Proceedings 17th International Conference on Data Engineering, 2001, p. 321-9, 23 refs, pp. xxii+666,
    ISBN: 0-7695-1001-9.
    Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.
Author(s)
    Schmidt-A, Kersten-M, Windhouwer-M.
Author affiliation
    Schmidt, A., Kersten, M., Windhouwer, M., CWI, Amsterdam, Netherlands.
Abstract
    Due to the ubiquity and popularity of **XML,** users often are in the following situation: they want to
    query **XML documents** which contain potentially interesting information but they are unaware of the
    mark-up structure that is used. For example, it is easy to guess the contents of an **XML** bibliography
    file whereas the mark-up depends on the methodological, cultural and personal background of the
    author(s). None the less, it is this hierarchical structure that forms the basis of **XML** query languages.
    We exploit the **tree** structure of **XML documents** to equip users with a powerful tool, the meet
    operator that lets them query databases with whose content they are familiar, but without requiring
    knowledge of **tags** and hierarchies. Our approach is based on computing the lowest common ancestor
    of nodes in the **XML** syntax **tree:** e.g., given two strings, we are looking for nodes whose offspring
    contains these two strings. The novelty of this approach is that the result type is unknown at query
    formulation time and dependent on the database instance. If the two strings are an author's name and
    a year mainly publications of the author in this year are returned. If the two strings are numbers the
    result mostly consists of publications that have the numbers as year or page numbers. Because the
    result type of a query is not specified by the user we refer to the lowest common ancestor as nearest
    concept. We also present a running example taken from the bibliography domain, and demonstrate

that the operator can be implemented efficiently.
Descriptors
     DATA-MODELS;   HYPERMEDIA-MARKUP-LANGUAGES;   MULTIMEDIA-DATABASES;
     QUERY-LANGUAGES;   QUERY-PROCESSING;   TREE-DATA-STRUCTURES.
Classification codes
     C6160M Multimedia-databases*;
     C6140D High-level-languages;
     C6130D Document-processing-techniques.
Keywords
     XML-document-querying; nearest-concept-queries; mark-up-structure; **XML-bibliography-file;**
     query-languages; **tree-structure; syntax-tree;** query-formulation-time.
Treatment codes
     P Practical.
Language
     English.
Publication type
     Conference-proceedings.
Availability
     CCCC: 1063-6382/2001/$10.00.
Digital object identifier
     10.1109/ICDE.2001.914844.
Publication year
     2001.
Publication date
     20010000.
Edition
     2001017.
Copyright statement
     Copyright 2001 IEE.

☑ **document 14 of 15** Order Document
Inspec - 1898 to date (INZZ)

Accession number & update
     0006663111 20051201.
Title
     An automated approach for retrieving hierarchical data from HTML tables.
Conference information
     Proceedings of CIKM99: Conference on Information and Knowledge Management, Kansas City, MI,
     USA, 2-6 Nov. 1999.
     Sponsor(s): ACM.
Source
     Proceedings of the Eighth International Conference on Information Knowledge Management. CIKM'99,
     1999, p. 466-74, 10 refs, pp. x+553, ISBN: 1-58113-146-1.
     Publisher: ACM, New York, NY, USA.
Author(s)
     Seung-Jin-Lim, Yiu-Kai-Ng.
     Editor(s): Gauch-S.
Author affiliation
     Seung-Jin Lim, Yiu-Kai Ng, Dept. of Comput. Sci., Brigham Young Univ., Provo, UT, USA.
Abstract
     Among the HTML elements, HTML tables encapsulate hierarchically structured data (hierarchical data in
     short) in a tabular structure. HTML tables do not come with a rigid schema and almost any forms of
     two-dimensional tables are acceptable according to the HTML grammar. This relaxation complicates the

process of retrieving hierarchical data from HTML tables. We propose an automated approach for retrieving hierarchical data from HTML tables. The proposed approach constructs the content **tree** of an HTML table, which captures the intended hierarchy of the data content of the table, without requiring the internal structure of the table be known beforehand. Also, the user of the content **tree** does not deal with HTML **tags** while retrieving the desired data from the content **tree.** Our approach can be employed by: (i) a query language written for retrieving hierarchically structured data, extracted from either the contents of HTML tables or other sources; (ii) a processor for converting HTML tables to **XML documents;** and (iii) a data warehousing repository for collecting hierarchical data from HTML tables and storing materialized views of the tables. The time complexity of the proposed retrieval approach is proportional to the number of HTML elements in an HTML table.

Descriptors
COMPUTATIONAL-COMPLEXITY; DATA-ENCAPSULATION; DATA-WAREHOUSES; HYPERMEDIA-MARKUP-LANGUAGES; INFORMATION-RETRIEVAL; QUERY-LANGUAGES; TREE-DATA-STRUCTURES.

Classification codes
C6130D Document-processing-techniques*;
C7250 Information-storage-and-retrieval;
C6160Z Other-DBMS;
C6120 File-organisation.

Keywords
hierarchical-data-retrieval; HTML-tables; data-encapsulation; tabular-structure; two-dimensional-tables; **content-tree;** query-language; processor; **XML-documents;** data-warehousing-repository; materialized-view-storage; time-complexity.

Treatment codes
P Practical;
T Theoretical-or-mathematical.

Language
English.

Publication type
Conference-proceedings.

Availability
CCCC: 1 58113 146 1/99/0010...$5.00.

Publication year
1999.

Publication date
19990000.

Edition
2000030.

Copyright statement
Copyright 2000 IEE.

COPYRIGHT BY The IET, Stevenage, UK

☑ **document 15 of 15** Order Document
Inspec - 1898 to date (INZZ)

Accession number & update
0006170834 20051201.

Title
An **XML document** to JavaScript object converter.

Source
WEB Techniques, {WEB-Tech-USA}, Jan. 1999, vol. 4, no. 1, p. 63-9, 0 refs, CODEN: WETEFA, ISSN: 1086-556X.
Publisher: Miller Freeman, USA.

Author(s)
Hildyard-A.

Abstract

XML is fast gaining currency as the standard for Web based data transmission. But how will **XML documents** be viewed by all those non **XML** browsers? The author has come up with an approach that brings some of the benefits of **XML** based **documents** to non **XML** browsers. His workaround is a server side conversion of **XML documents** to JavaScript code; this code gets interpreted by the browser and results in a data structure roughly equivalent to the parse **tree** that would have been produced by an **XML** enabled browser. Transforming **XML documents** from **tag** stream to DOM **(Document** Object Model) provides a similar benefit of increased accessibility for the data consumer that moving data from databases to **XML** data sources provides for data producers. With **XML** represented at the level of the DOM, Web based consumers are freed from both the need for an **XML** parser and also from the need to have direct access to original **XML** data sources. Applets, scriptlets, ActiveX controls, and other client side components have the same programmatic access to browser based **XML documents** as they have to the rest of the browser's DOM. As it turns out, this workaround offers significant advantages over a pure **XML** approach: it's a lot faster, and the code to manipulate **XML** derived objects is cleaner and more concise.

Descriptors

AUTHORING-LANGUAGES; CLIENT-SERVER-SYSTEMS; DISTRIBUTED-OBJECT-MANAGEMENT; HYPERMEDIA-MARKUP-LANGUAGES; INTERNET; JAVA; ONLINE-FRONT-ENDS.

Classification codes

C6130D Document-processing-techniques*;
C6130M Multimedia;
C6140D High-level-languages;
C6110J Object-oriented-programming;
C6150N Distributed-systems-software;
C6115 Programming-support;
C7250N Search-engines;
C7210N Information-networks.

Keywords

XML-document; JavaScript-object-conversion; Web-based-data- transmission; **non-XML-browsers;** workaround; server-side-conversion; JavaScript-code; data-structure; **parse-tree; XML-enabled-browser; tag-** stream; DOM; **Document-Object-Model;** data-consumer; **XML-data-sources;** Web-based-consumers; applets; scriptlets; ActiveX-controls; client-side-components; programmatic-access; **browser-based-XML-documents; XML-derived-objects.**

Treatment codes

P. Practical.

Language

English.

Publication type

Journal-paper.

Availability

SICI: 1086-556X(199901)4:1L.63:DJOC; 1-R.

Publication year
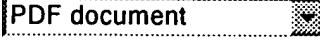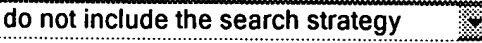
1999.

Publication date

19990100.

Edition

1999008.

Copyright statement

Copyright 1999 IEE.

locally as: | PDF document ▾ | search strategy: | do not include the search strategy ▾ |